Preconditioning the Helmholtz equation via row-projections

Tristan van Leeuwen[1], Dan Gordon[2], Rachel Gordon[3] and Felix J. Herrmann[1]
[1] Dept. of Earth and Ocean Sciences, University of British Columbia, 6339 Stores rd, Vancouver, BC, V6T1Z4
[2] Dept. of Computer Science, University of Haifa, Haifa 31905, Israel
[3] Dept. of Aerospace Engineering, The Technion, Haifa 32000, Israel

January 17, 2012

Abstract

3D frequency-domain full waveform inversion relies on being able to efficiently solve the 3D Helmholtz equation. Iterative methods require sophisticated preconditioners because the Helmholtz matrix is typically indefinite. We review a preconditioning technique that is based on row-projections. Notable advantages of this preconditioner over existing ones are that it has low algorithmic complexity, is easily parallelizable and extendable to time-harmonic vector equations.

3D frequency-domain full waveform inversion relies on being able to efficiently solve the 3D Helmholtz equation. A direct factorization of the matrix –as is commonly done for 2D waveform inversion– is infeasible because of the huge memory requirements, although some progress has been recently made in this area (Wang et al., 2011). Iterative methods have a small memory imprint, but require sophisticated preconditioners because the matrix is indefinite. Over the past years, several preconditioners have been proposed in the literature, and we briefly discuss a few. A good overview of classical preconditioning techniques, such as ILU and Schwarz is given by Osei-Kuffuor and Saad (2010). Sourbier et al. (2011) propose a hybrid method based on these techniques. Preconditioners based on the shifted Laplacian, first proposed by Bayliss et al. (1983), were further developed by Erlangga et al. (2004) who use a multigrid method in conjunction with a complex shift; Riyanti et al. (2007) discuss a parallelization. Stolk (2010); Haber and MacLachlan (2011) propose preconditioners that are based on asymptotic solutions of the Helmholtz equation. Recently, Engquist (2011) proposed a 'sweeping' preconditioner. All these methods are generally used in conjunction with iterative methods like GMRES or Bi-CG(STAB).
In this paper we review a preconditioning technique that is particularly appealing because of its simplicity. The main idea is that any system of equations can be transformed into an equivalent, positive semi-definite system via KACZMARZ row-projections. The resulting system can then be solved efficiently by the Conjugate Gradient (CG) algorithm. The original method is due to Björck and Elfving (1979), who dubbed the method CGMN. The problem with this approach is that KACZMARZ is inherently sequential. Gordon and Gordon (2005) proposed the CARP algorithm, which is a domain decomposition-based approach to parallelizing KACZMARZ, with data transfer between processors occurring at the subdomain boundaries only. Basically, CARP operates by first partitioning the system of equations into blocks (which may overlap) and repeats the following two operations until convergence: *i)* KACZMARZ row projections are applied to the equations in each block, independently and in parallel; *ii)* any variable shared by two or more blocks is updated to be the average of its values in the different blocks. Gordon and Gordon (2010) propose a CG acceleration of CARP, which can be seen as a parallel extension of CGMN. CARP-CG was shown to be particularly useful for problems in which the system matrix has very large off-diagonal elements, including cases with discontinuous coefficients; see Gordon and Gordon (2009, 2010). The main advantages of this preconditioner over existing ones are: *i)* the preconditioner is independent of the input matrix; *ii)* the algorithm is embarassingly parallel, and *iii)* The costs per iteration are very low.
The paper is organized as follows. First, we briefly review the CARP-CG algorithm. We present some numerical experiments in 1D to study the behaviour of the preconditioned system in terms of the eigenvalues. Numerical examples on the 3D SEG/EAGE salt model show how CARP-CG scales with frequency. Finally, we draw conclusions and point out directions for future research.

The CARP-CG algorithm

The Kaczmarz method solves a system of equations, $A\mathbf{u} = \mathbf{b}$, by cyclically projecting the iterate onto rows of the matrix (Kaczmarz, 1937). One sweep through the whole matrix is then given by:

$$\mathbf{u} := \mathbf{u} + \omega_i \left(b_i - \mathbf{a}_i{}^T \mathbf{u}\right) \mathbf{a}_i / ||\mathbf{a}_i||_2^2, \quad i = 0 \ldots N-1, \tag{1}$$

where $\mathbf{a}_i$ denotes the i-th row of A as column vector, $0 < \omega_i < 2$ is a relaxation parameter and N denotes the number or rows in the matrix. Introducing the matrices $Q_i = \left(I - \omega_i \mathbf{a}_i \mathbf{a}_i{}^T / ||\mathbf{a}_i||_2^2\right)$, we may write $\mathbf{u} := Q_i\mathbf{u} + \omega_i b_i \mathbf{a}_i / ||\mathbf{a}_i||_2^2$. A forward sweep followed by a backward sweep (hereafter referred to as a double sweep) yields: $\mathbf{u} := Q\mathbf{u} + R\mathbf{b}$, where $Q = Q_0 Q_1 \ldots Q_{N-1} Q_{N-1} \ldots Q_0$ and R contains all the factors multiplying $\mathbf{b}$. The Q_i are all symmetric rank 1 matrices with eigenvalues $1 - \omega_i$. Hence, Q is symmetric and has eigenvalues $\in [-1, 1]$. We may now transform the original system of equations to a symmetric positive semi-definite system: $(I - Q)\mathbf{u} = R\mathbf{b}$.
We never have to explicitly form the matrices Q and R. Instead, we can calculate the action of these matrices using Algorithm 1 as $(I - Q)\mathbf{u} + R\mathbf{b} = \mathsf{DKSWP}(A, \mathbf{u}, \mathbf{b}, \omega)$. The CARP-CG algorithm applies CG to solve this equivalent system.
For a k-point stencil, a matrix-vector multiplication with the matrix $I - Q$ via the DKSWP algorithm

requires $(4k+2)N$ flops (compared to kN for the original matrix A).
In practice, we decompose the domain and perform the row-projections for the blocks in parallel. After a double sweep the gridpoints near the boundary are communicated and averaged. A detailed description is given in Gordon and Gordon (2010).

Numerical study in 1D

We study the effects of the discussed preconditioning technique on the 1D Helmholtz equation on the domain $[0,1]$. We use a 3-point stencil with Dirichlet boundary conditions. The wavenumber is given by $k(x) = k_0 p(x)$. The profile p is depicted in figure 1 (a). The gridspacing is varied with k_0 to ensure a minimum of 10 gridpoints per wavelength. The solution for k_0 = 100 and a pointsource at $x = \frac{1}{2}$ is depicted in figure 1 (b). We compare solving the following three equivalent systems with CG: *i)* The normal equations: $A^T A\mathbf{u} = A^T\mathbf{b}$; *ii)* the normalized normal equations: $A^T W^T WA\mathbf{u} = A^T W^T W\mathbf{b}$, where $W = \mathrm{diag}(\mathbf{w}), w_i = 1/||\mathbf{a}_i||_2$ and *iii)* the preconditioned system $(I-Q)\mathbf{u} = R\mathbf{b}$ for various values of the relaxation parameter ω. The eigenvalues of all three matrices are depicted in figure 1 (c). The eigenvalues of $(I-Q)$ are increasingly more clustered and move away from zero for larger values of the relaxation parameter, which is advantageous for iterative solution methods. The number of CG iterations needed to reach a fixed tolerance for various k_0 is shown in figure 1 (d). The preconditioner significantly lowers the number of iterations needed. However, the number of iterations still increases with the wavenumber.

Numerical example in 3D

We calculate wavefields in the SEG/EAGE salt model, depicted in figure 2 (a). The Helmholtz equation is discretized using a 7-point discretization with first order absorbing boundary conditions with grid spacings of h = 20, 40, 80 and 160 m. We use a point source in the center of the $z = 0$ plane. The number of complex variables variesfrom 195,075 to 95,964,960. The total domain is decomposed in 12 horizontal layers and we use a fixed relaxation parameter $\omega = 1.5$. All the results are calculated using a C/MPICH2 implementation of the CARP-CG algorithm on a 12-node SuperMicro cluster with two Intel Xeon E5520 quad CPUs running at 2.27 GHz and 8 GB memory per node, connected by an Infiniband network. For the experiments we used only one core per node. The gridspacing is varied with frequency to ensure a minimum of 7.5 gridpoints per wavelength. The wavefield at 5 Hz is depicted in figure 2 (b). The convergence histories for different frequencies are depicted in figure 3. Note that the convergence is extremely regular. Iteration counts and timings for different frequencies and tolerances are shown in table 1.

Conclusion

CARP-preconditioning can substantially reduce the number of CG iterations needed compared to solving the normal equations. While this can still lead to a large number of iterations, the iterations themselves are very cheap. Success of this approach in practice hinges on an efficient implementation of the algorithm, since CPU time is the critical factor in practice, not the number of iterations.
The advantage of this approach over existing multi-grid-based approaches are: *i)* its algorithmic simplicity, *ii)* its parallelism and good scalability and *iii)* its independence of the input matrix. The latter implies that we do not need to re-design the preconditioner if we change the physics (e.g., use an elastic wave equation). Preliminary bechmarking results (Gordon and Gordon, 2012) suggest that CARP-CG can outperform a multi-grid preconditioner based on the shifted Laplacian.
Future research will be aimed at studying the influence of the relaxation parameter and the choice of domain layout as well as comparing CARP-CG to other preconditioners. Higher order schemes will also be considered. To facilitate benchmarking, the code used for the 3D example can be downloaded from http://cs.haifa.ac.il/ gordon/soft.html.

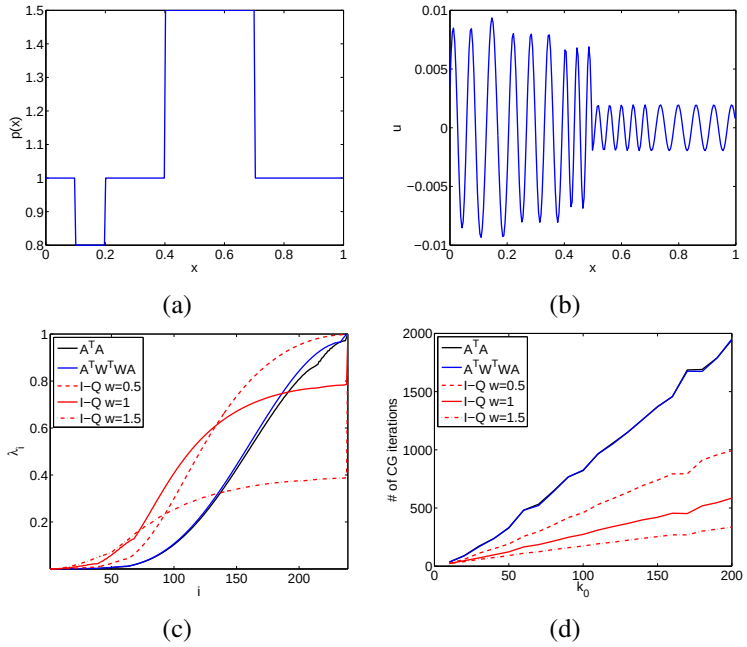

Figure 1 *(a) wavenumber $k(x) = 100p(x)$, (b) solution $u(x)$, (c) eigenvalues of corresponding matrices. The eigenvalues of the normal equations and the normalized normal equations are divided by their maximum for display purposes. (d) number of CG iterations required to reach a given tolerance.*

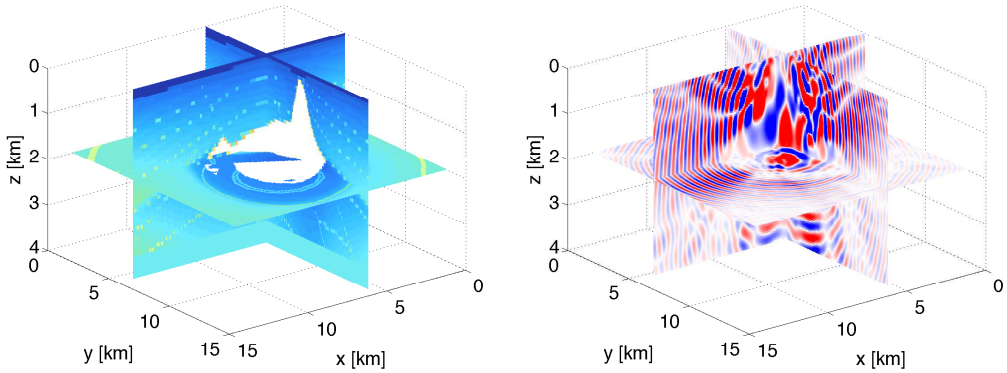

Figure 2 *velocity model and wavefield at 5 hz.*

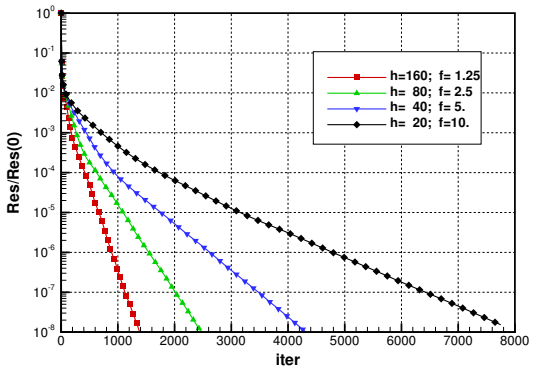

Figure 3 *Convergence history for different frequencies for a fixed number of gridpoints per wavelength.*

		$\varepsilon = 10^{-3}$		$\varepsilon = 10^{-5}$	
f	N	iterations	time (s)	iterations	time (s)
1.25	195075	180	1.34	666	4.98
2.5	1513733	267	12.4	1092	50
5	11995620	443	118.7	1758	471
10	95964960	709	1418	3157	6314

Table 1 *Iteration counts and timings for different frequencies and tolerances for a minimum of 7.5 gridpoints per wavelength. N denotes the total number of complex variables.*

Algorithm 1 DKSWP$(A, \mathbf{u}, \mathbf{b}, \omega)$

forward sweep
for $i = 0$ to $N-1$ **do**
  $\mathbf{u} := \mathbf{u} + \omega_i(b_i - \mathbf{a}_i{}^T\mathbf{u})\mathbf{a}_i/||\mathbf{a}_i||_2^2$
end for
backward sweep
for $i = N-1$ to 0 **do**
  $\mathbf{u} := \mathbf{u} + \omega_i(b_i - \mathbf{a}_i{}^T\mathbf{u})\mathbf{a}_i/||\mathbf{a}_i||_2^2$
end for
return u

Acknowledgments

The authors thank Martin Pajchel who did an internship with the SLIM group at UBC. This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, Chevron, ConocoPhillips, Petrobras, PGS, Total SA, and WesternGeco.

References

Bayliss, A., Goldstein, C.I. and Turkel, E. [1983] An iterative method for the Helmholtz equation. *Journal of Computational Physics*, **49**, 443–457.

Björck, Å. and Elfving, T. [1979] Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations. *BIT*, **19**, 145–163.

Engquist, B. [2011] Sweeping Preconditioner for the Helmholtz Equation : Hierarchical Matrix Representation. *Communications on pure and applied mathematics*, **LXIV**, 0697–0735.

Erlangga, Y.A., Vuik, C. and Oosterlee, C.W. [2004] On a class of preconditioners for solving the Helmholtz equation. *Applied Numerical Mathematics*, **50**, 409–425.

Gordon, D. and Gordon, R. [2005] Component-averaged row projections: A robust, block-parallel scheme for sparse linear systems. *SIAM J. on Scientific Computing*, **27**, 1092–1117.

Gordon, D. and Gordon, R. [2009] Solution methods for linear systems with large off-diagonal elements and discontinuous coefficients. *Computer Modeling in Engineering & Sciences*, **53**(1), 23–45.

Gordon, D. and Gordon, R. [2010] Carp-cg: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated pdes. *Parallel Computing*, **36**, 495–515.

Gordon, D. and Gordon, R. [2012] Robust and highly scalable parallel solution of the helmholtz equation with large wave numbers. *Tech. Report*.

Haber, E. and MacLachlan, S. [2011] A fast method for the solution of the Helmholtz equation. *Journal of Computational Physics*, 1–28.

Kaczmarz, S. [1937] Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, **35**, 355–357.

Osei-Kuffuor, D. and Saad, Y. [2010] Preconditioning Helmholtz linear systems. *Applied numerical mathematics*, **60**(4), 420–431.

Riyanti, C. et al. [2007] A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. *Journal of Computational Physics*, **224**(1), 431–448.

Sourbier, F., Haidar, A., Giraud, L., Ben-Hadj-Ali, H., Virieux, J. and Operto, S. [2011] Three-dimensional parallel frequency-domain visco-acoustic wave modelling based on a hybrid direct / iterative solver. *Geophysical Prospecting*, **59**, 834–856.

Stolk, C. [2010] On the generalization of wavelet diagonal preconditioning to the Helmholtz equation. *Arxiv preprint arXiv:1010.4764*, 1–18.

Wang, S., de Hoop, M. and Xia, J. [2011] On 3D modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver. *Geophysical Prospecting*, **59**(5), 857–873.